

Конспект по книге Гудфеллоу «Глубокое обучение»*

Содержание

1 Численные методы	1
2 Основы машинного обучения	1
2.1 Точечная оценка	1
2.2 Смещение	2
2.3 Дисперсия	2
2.4 Поиск компромисса между смещением и дисперсией для минимизации среднеквадратической ошибки	2
2.5 Состоятельность	3
2.6 Оценка максимального правдоподобия	3
2.7 Метод опорных векторов	4
2.8 Метод главных компонент	5
2.9 Стохастический градиентный спуск	6
2.10 Условное логарифмическое правдоподобие и среднеквадратическая ошибка	6
3 Глубокие сети прямого распространения	7
3.1 Обучение условных распределений с помощью максимального правдоподобия	7
3.2 Сигмоидные блоки и выходное распределение Бернулли	7
Список литературы	8

1. Численные методы

2. Основы машинного обучения

2.1. Точечная оценка

Точечное оценивание – это попытка найти единственное «наилучшее» предсказание интересующей величины. Пусть $\{x^{(1)}, \dots, x^{(m)}\}$ – множество m независимых и одинаково распределенных точек. *Точечной оценкой*, или *статистикой*, называется любая функция этих данных

$$\theta_m = g(x^{(1)}, \dots, x^{(m)}).$$

В этом определении не требуется, чтобы g возвращала значение, близкое к истинному значению θ , ни даже чтобы область значений g совпадала со множеством допустимых значений θ .

Алгоритм k -групповой перекрестной проверки применяется для оценивания ошибки обобщения алгоритма обучения A , когда имеющийся набор данных \mathbb{D} *слишком мал* для того, чтобы

*Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. – М.: ДМК Пресс, 2018. – 652 с.

простое разделение на обучающий и тестовый или обучающий и контрольный наборы могло дать точную оценку ошибки обобщения, поскольку среднее значение потери L на малом тестовом наборе может иметь высокую дисперсию.

2.2. Смещение

Смещение оценки определяется следующим образом

$$\text{bias}(\hat{\theta}_m) = \mathbb{E}(\hat{\theta}_m) - \theta,$$

где математическое ожидание вычисляется по данным (рассматривается как выборка из случайной величины), а θ – истинное значение параметра, которое определяет порождающее распределение.

Оценка $\hat{\theta}$ называется *несмещенной*, если

$$\text{bias}(\hat{\theta}_m) = 0, \text{ т.е. } \mathbb{E}(\hat{\theta}_m) = \theta.$$

Оценка $\hat{\theta}_m$ называется *асимптотически несмещенной*, если

$$\lim_{m \rightarrow \infty} \text{bias}(\hat{\theta}_m) = 0, \text{ т.е. } \lim_{m \rightarrow \infty} \mathbb{E}(\hat{\theta}_m) = \theta.$$

2.3. Дисперсия

Для определения смещения мы вычисляли математическое ожидание оценки, но точно так же можем вычислить и ее дисперсию. *Дисперсией оценки* называется выражение

$$\text{Var}(\hat{\theta}).$$

Стандартной ошибкой $\text{SE}(\hat{\theta})$ называется квадратный корень из дисперсии.

Воспользовавшись центральной предельной теоремой, согласно которой среднее имеет приблизительно нормальное распределение, можем применить стандартную ошибку для вычисления вероятности того, что истинное математическое ожидание находится в выбранном интервале. Например, *95-процентный доверительный интервал* вокруг выборочного среднего (вокруг оценки) $\hat{\mu}_m = \frac{1}{m} \sum_{k=1}^n x^{(i)}$ определяется формулой

$$(\hat{\mu}_m - 1.96 \text{SE}(\hat{\mu}_m), \hat{\mu}_m + 1.96 \text{SE}(\hat{\mu}_m))$$

при нормальном распределении со средним $\hat{\mu}_m$ и дисперсией $\text{SE}(\hat{\mu}_m)^2$.

NB: В экспериментах по машинному обучению принято говорить, что алгоритм A лучше алгоритма B , если верхняя граница 95-процентного доверительного интервала для ошибки алгоритма A меньше нижней границы 95-процентного доверительного интервала для ошибки алгоритма B .

2.4. Поиск компромисса между смещением и дисперсией для минимизации среднеквадратической ошибки

Что, если имеются две оценки, у одной из которых больше смещение, а у другой дисперсия? Какую выбрать?

Самый распространенный подход к выбору компромиссного решения – воспользоваться *перекрестной проверкой*. Эмпирически продемонстрировано, что перекрестная проверка дает отличные результаты во многих реальных задачах.

Можно также сравнить среднеквадратическую ошибку (MSE) обеих оценок

$$\text{MSE} = \mathbb{E}[(\hat{\theta}_m - \theta)^2] = \text{bias}(\hat{\theta}_m)^2 + \text{Var}(\hat{\theta}_m)$$

Желательной является оценка с малой MSE, именно такие оценки держат под контролем и смещение, и дисперсию. Соотношение между смещением и дисперсией тесно связано с возникающими в машинном обучении понятиями емкости модели, недообучения и переобучения.

Если ошибка обобщения измеряется посредством MSE (и тогда смещение и дисперсия становятся важными компонентами ошибки обобщения), то увеличение емкости (то есть *усложнение модели*) влечет за собой *повышение дисперсии* и *снижение смещения*.

2.5. Состоятельность

Обычно нас интересует также поведение оценки по мере роста размера обучающего набора. В частности, мы хотим, чтобы при увеличении числа примеров точечные оценки сходились к истинным значениям соответствующих параметров.

Формально это записывается в виде (условие состоятельности)

$$\hat{\theta}_m \xrightarrow{\mathbf{P}} \theta, \quad (m \rightarrow \infty)$$

Иногда это условие называют *слабой состоятельностью*, понимая под *сильной состоятельностью* сходимость *почти наверное* $\hat{\theta}$ к θ .

Состоятельность гарантирует, что смещение оценки уменьшается с ростом числа примеров. Однако обратное неверно – *из асимптотической несмещенности не вытекает состоятельность*. Рассмотрим, к примеру, оценивание среднего μ нормального распределения $N(x; \mu, \sigma^2)$ по набору данных, содержащему m примеров: $\{x^{(1)}, \dots, x^{(m)}\}$.

Можно было бы взять в качестве оценки первый пример: $\hat{\theta} = x^{(i)}$. В таком случае $\mathbb{E}(\hat{\theta})_m = \theta$, поэтому оценка является несмещенной вне зависимости от того, сколько примеров мы видели. Отсюда, конечно, следует, что оценка асимптотически несмещенная. Но она не является состоятельной, т.к. *неверно*, что $\hat{\theta}_m \rightarrow \theta, (m \rightarrow \infty)$.

2.6. Оценка максимального правдоподобия

Рассмотрим множества m примеров $\mathbb{X} = \{x^{(1)}, \dots, x^{(m)}\}$, независимо выбираемых из неизвестного порождающего распределения $p_{data}(x)$.

Обозначим $p_{model}(x; \theta)$ параметрическое семейство распределений вероятности над одним и тем же пространством, индексированное параметром θ .

Тогда оценка максимального правдоподобия для θ определяется формулой

$$\theta_{ML} = \arg \max_{\theta} p_{model}(\mathbb{X}; \theta) = \arg \max_{\theta} \prod_{i=1}^m p_{model}(x^{(i)}; \theta)$$

Такое произведение большого числа вероятностей по ряду причин может быть неудобно. Например, оно подвержено *потере значимости*. Для получения эквивалентной, но более удобной

задачи оптимизации заметим, что взятие логарифма правдоподобия не изменяет $\arg \max$, но преобразует произведение в сумму

$$\theta_{ML} = \arg \max_{\theta} \sum_{i=1}^m \log p_{model}(x^{(i)}; \theta)$$

Поскольку $\arg \max$ не изменяется при умножении функции стоимости на константу, мы можем разделить правую часть на m и получить выражение в виде математического ожидания относительно эмпирического распределения \hat{p}_{data} , определяемого обучающими данными

$$\theta_{ML} = \arg \max_{\theta} \mathbb{E}_{x \sim \hat{p}_{data}} [\log p_{model}(x; \theta)]$$

Один из способов интерпретации оценки максимального правдоподобия состоит в том, чтобы рассматривать ее как минимизацию дивергенции (расхождения) Кульбака-Лейблера между этими эмпирическим распределением \hat{p}_{data} , определяемым обучающим набором, и модельным распределением.

Дивергенция Кульбака-Лейблера определяется формулой

$$D_{KL}(\hat{p}_{data} || p_{model}) = \mathbb{E}_{x \sim \hat{p}_{data}} [\log \hat{p}_{data}(x) - \log p_{model}(x)]$$

Первый член разности в квадратных скобках зависит только от порождающего данные процесса, но не от модели. Следовательно, при обучении модели, минимизирующей дивергенцию КЛ, мы должны минимизировать только величину

$$-\mathbb{E}_{x \sim \hat{p}_{data}} [\log p_{model}(x)],$$

а это, конечно, то же самое, что максимизация величины $\theta_{ML} = \arg \max_{\theta} \mathbb{E}_{x \sim \hat{p}_{data}} [\log p_{model}(x; \theta)]$.

НВ: То есть, другими словами задача максимизации правдоподобия эквивалентна задаче минимизации дивергенции Кульбака-Лейблера между эмпирическим распределением \hat{p}_{data} и модельным распределением p_{model} .

2.7. Метод опорных векторов

Линейную функцию в методе опорных векторов можно переписать в виде

$$w^T x + b = b + \sum_{i=1}^m \alpha_i x^T x^{(i)},$$

где $x^{(i)}$ – обучающий пример, α – вектор коэффициентов.

Записав алгоритм обучения в таком виде, мы сможем заменить x результатом заданной функции признаков $\varphi(x)$, а скалярное произведение – функцией $k(x, x^{(i)}) = \varphi(x) \cdot \varphi(x^{(i)})$, которая называется ядром.

Заменив скалярное произведение вычислением ядра, мы можем делать предсказание, пользуясь функцией

$$f(x) = b + \sum_i \alpha_i k(x, x^{(i)})$$

Основанная на ядре функция в точности эквивалентна предварительной обработке путем применения $\varphi(x)$ ко всем входным данным с последующим обучением линейной модели в новом преобразованном пространстве.

NB: Трюк с ядром полезен по двум причинам:

- Во-первых, он позволяет обучать модели, *нелинейно* зависящие от x , применяя методы выпуклой оптимизации, о которых точно известно, что они сходятся эффективно
- Во-вторых, *ядерная функция k* часто допускает реализацию, значительно *более эффективную с вычислительной точки зрения*, чем наивное построение двух векторов $\varphi(x)$ и явное вычисление их скалярного произведения

Главный недостаток ядерных методов – тот факт, что сложность вычисления решающей функции линейно зависит от числа обучающих примеров, поскольку i -ый пример вносит член $\alpha_i k(x, x^{(i)})$ в решающую функцию.

В методе опорных векторов эта проблема сглаживается тем, что обучаемый вектор α содержит в основном нули. *Тогда для классификации нового примера требуется вычислить ядерную функцию только для обучающих примеров с ненулевыми α_i .* Эти обучающие примеры и называются опорными векторами.

2.8. Метод главных компонент

Метод главных компонент находит ортогональное линейное преобразование, переводящее входные данные x в представление z .

Рассмотрим матрицу плана X размера $m \times n$. Будем предполагать, что математическое ожидание данных $\mathbb{E}[x] = 0$. Если это не так, центрирования легко добиться, вычтя среднее из всех примеров на этапе предварительной обработки.

Несмещенная выборочная ковариационная матрица, ассоциированная с X , определяется по формуле

$$\text{Var}[x] = \frac{1}{m-1} X^T X$$

PCA находит представление (посредством линейного преобразования) $z = W^T x$, для которого $\text{Var}[z]$ – *диагональная*.

Главные компоненты можно получить с помощью сингулярного разложения. Точнее, это правые сингулярные векторы. Чтобы убедиться в этом, предположим, что W – правые сингулярные векторы в разложении $X = U \Sigma W^T$. Тогда исходное уравнение собственных векторов можно переписать в базисе W

$$X^T X = (U \Sigma W^T)^T U \Sigma W^T = W \Sigma^2 W^T$$

Разложение SVD полезно для доказательства того, что PCA приводит к диагональной матрице $\text{Var}[z]$. Применяя сингулярное разложение X , мы можем выразить дисперсию X в виде

$$\text{Var}[x] = \frac{1}{m-1} X^T X = \frac{1}{m-1} (U \Sigma^2 W^T)^T U \Sigma W^T = \frac{1}{m-1} W \Sigma^2 W^T,$$

где используется тот факт, что $U^T U = I$, поскольку матрица U в сингулярном разложении по определению ортогональная. Отсюда следует, что ковариационная матрица z диагональная

$$\text{Var}[z] = \frac{1}{m-1} Z^T Z = \frac{1}{m-1} W^T X^T X W = \frac{1}{m-1} W^T W \Sigma^2 W^T W = \frac{1}{m-1} \Sigma^2$$

На этот раз мы воспользовались тем, что $W^T W = I$ – опять же по определению сингулярного разложения.

Проведенный анализ показывает, что *представление, полученное в результате проецирования данных x на z посредством линейного преобразования W , имеет диагональную ковариационную матрицу Σ^2 . А отсюда сразу вытекает, что взаимная корреляция отдельных элементов z равна нулю.*

2.9. Стохастический градиентный спуск

Стохастический градиентный спуск (СГС) имеет важные применения и за пределами глубокого обучения. Это основной способ обучения *большим линейным моделям* на очень больших наборах данных. Для модели фиксированного размера стоимость одного шага СГС не зависит от размера обучающего набора m . Количество шагов до достижения сходимости обычно возрастает с ростом размера обучающего набора. Но когда m стремится к бесконечности, модель в итоге сходится к наилучшей возможной ошибке тестирования еще до того, как СГС проверил каждый пример из обучающего набора. Дальнейшее увеличение m не увеличивает время обучения, необходимое для достижения наилучшей ошибки тестирования. *С этой точки зрения можно сказать, что асимптотическая стоимость обучения модели методом СГС как функции от m имеет порядок $O(1)$.*

2.10. Условное логарифмическое правдоподобие и среднеквадратическая ошибка

Линейную регрессию можно интерпретировать как *нахождение оценки максимального правдоподобия*. Будем считать, что цель не в том, чтобы вернуть одно предсказание \hat{y} , а чтобы построить модель, порождающую условное распределение $p(y|\mathbf{x})$. Цель алгоритма обучения теперь – аппроксимировать $p(y|\mathbf{x})$, подогнав его под все эти разные значения y , совместимые с \mathbf{x} .

Для вывода такого же алгоритма линейной регрессии, как и раньше, определим

$$p(y|\mathbf{x}) = N(y; \hat{y}(\mathbf{x}, \mathbf{w}), \sigma^2)$$

Функция $\hat{y}(\mathbf{x}; \mathbf{w})$ дает предсказание среднего значения нормального распределения. В этом примере мы предполагаем, что дисперсия фиксирована и равна константе σ^2 . Поскольку предполагается, что примеры независимы и одинаково распределены, то условное логарифмическое правдоподобие записывается в виде

$$\sum_{i=1}^m \log p(y^{(i)}|\mathbf{x}^{(i)}; \theta) = -m \log \sigma - \frac{m}{2} \log 2\pi - \sum_{i=1}^m \frac{\|\hat{y}^{(i)} - y^{(i)}\|^2}{2\sigma^2},$$

где $\hat{y}^{(i)}$ – результат линейной регрессии для i -ого примера $\mathbf{x}^{(i)}$, а m – число обучающих примеров.

Сравнивая логарифмическое правдоподобие со среднеквадратической ошибкой

$$\text{MSE}_{\text{train}} = \frac{1}{m} \sum_{i=1}^m \|\hat{y}^{(i)} - y^{(i)}\|^2,$$

мы сразу же видим, что *максимизация логарифмического правдоподобия* относительно \mathbf{w} дает ту же оценку параметров \mathbf{w} , что *минимизация среднеквадратической ошибки*.

Значения этих критериев различны, но положение оптимума совпадает. Это служит обоснованием использования среднеквадратической ошибки в качестве оценки максимального правдоподобия.

3. Глубокие сети прямого распространения

3.1. Обучение условных распределений с помощью максимального правдоподобия

Большинство современных нейронных сетей обучается *с помощью максимального правдоподобия*. Это означает, что *в качестве функции стоимости берется отрицательное логарифмическое правдоподобие*, которое можно эквивалентно описать как перекрестную энтропию между обучающими данными и распределением модели

$$J(\theta) = -\mathbb{E}_{\mathbf{x}, y \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{y}|\mathbf{x})$$

Одно необычное свойство перекрестной энтропии, используемой при вычислении оценки максимального правдоподобия, заключается в том, что для типичных встречающихся на практике моделей у нее, как правило, нет минимального значения. Если выходная величина дискретна, то в большинстве моделей параметризация устроена так, что модель неспособна представить вероятность 0 или 1, но может подойти к ней сколь угодно близко. Примером может служить логистическая регрессия.

3.2. Сигмоидные блоки и выходное распределение Бернулли

Во многих задачах требуется предсказывать значение бинарной величины y . В таком виде можно представить задачу классификации с двумя классами.

Подход на основе максимального правдоподобия заключается в определении распределения Бернулли величины y при условии \mathbf{x} .

Градиент 0 обычно приводит к проблемам, потому что у алгоритма обучения нет никаких указаний на то, как улучшить параметры.

Лучше применять другой подход, который гарантирует, что градиент обязательно будет достаточно большим, если модель дает неверный ответ. Этот подход основан на использовании сигмоидных выходных блоков в сочетании с максимальным правдоподобием.

Сигмоидный выходной блок

$$\hat{y} = \sigma(\mathbf{w}^T \mathbf{h} + b),$$

где σ – логистическая сигмоида.

Список литературы

1. *Рамальо Л.* Python – к вершинам мастерства: Лаконичное и эффективное программирование. – М.: МК Пресс, 2022. – 898 с.
2. *Хейдт М., Груздев А.* Изучаем pandas. – М.: ДМК Пресс, 2019. – 682 с.